ParaCLEAN: Improving Translation Quality Through Systematic Parallel Data Cleaning

Audrey Mash*, **Ella Bohman***, **Maite Melero**
Barcelona Supercomputing Center
Barcelona, Spain
firstname.lastname@bsc.es

Abstract

Parallel corpora often contain significant noise, particularly in low-resource settings where both collected and synthetic data are combined. We present ParaCLEAN, a modular pipeline for cleaning parallel data that integrates embedding-based filtering, language identification, deduplication, and normalisation. Experiments on Catalan→Japanese demonstrate that ParaCLEAN improves data quality and downstream MT performance. Ablation studies highlight the contribution of each step. ParaCLEAN is lightweight, reproducible, and extensible for diverse language pairs.

Keywords: parallel corpora, data cleaning, low-resource MT, Catalan, Japanese

1. Introduction

The quality of parallel data is a decisive factor in the performance of machine translation systems. Large-scale collections such as OPUS and ELRC provide extensive resources, but they often contain noise in the form of misaligned sentences, language identification errors, duplicates, and inconsistent formatting. These issues can reduce system robustness, hinder evaluation, and complicate downstream applications.

To address this, we present ParaCLEAN, a modular pipeline for cleaning and normalising parallel corpora. ParaCLEAN has been designed with sentence-level data in mind, enabling fine-grained alignment, filtering, and quality assessment. It integrates existing tools and newly developed procedures into a reproducible workflow that can be applied across language pairs. The pipeline includes language identification, embedding-based filtering, deduplication, optional bilingual restorative cleaning, and language-specific normalisation. We release the full ParaCLEAN codebase and configuration files for reproducibility. [1]

We evaluate ParaCLEAN in the context of Catalan→Japanese machine translation, a low-resource and structurally diverse pair. This setting highlights the importance of data quality and allows us to measure the individual contributions of different cleaning steps.

2. Related Work

Data quality has long been recognized as a critical factor in the performance of machine translation and other multilingual NLP systems. Large-scale parallel corpora are known to contain substantial noise, including misaligned or untranslated sentences and incorrect language labels, with misalignment often the most damaging source of error (Khayrallah and Koehn, 2018). Large web-derived corpora such as CommonCrawl contain extensive noise and non-linguistic material (Luccioni and Viviano, 2021), and even widely used parallel datasets exhibit low proportions of acceptable translations, from roughly 31% in CCMatrix to 64% in MaCoCu (Van Noord et al., 2025). Although corpus size can partially offset low quality, quality remains a key determinant of performance when size is controlled. In addition, many large multilingual datasets include substantial portions of unusable content, with some languages represented by little or no valid data and "a significant fraction" of others containing less than 50% usable material (Kreutzer et al., 2022). These findings underscore the importance of systematic data cleaning and quality assessment prior to model training.

Several shared tasks at WMT (Koehn et al., 2018, 2019, 2020) have specifically addressed the problem of filtering noisy web-crawled data, leading to notable advances in data cleaning for machine translation. The highest performing MT systems in these tasks (Lu et al., 2020; Açarçiçek et al., 2020) generally rely on supervised filtering models, trained on clean, language-specific parallel data provided as part of the shared task resources. Although effective for well-resourced language pairs, this approach does not scale to massively multilingual or low-resource settings where such high-quality training data are unavailable. Similarly,

*Audrey Mash and Ella Bohman contributed equally to this work.

[1] https://github.com/langtech-bsc/ParaCLEAN

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 6630–6640
11–16 May 2026.

one of the most widely adopted filtering tools, Bicleaner (Ramírez-Sánchez et al., 2020; Zaragoza-Bernabeu et al., 2022), faces the same constraint: it requires parallel training data for each language pair and only provides pre-trained models for a small set of languages, all paired with English. Other frameworks, such as OpusFilter (Aulamo et al., 2020), offer more general and configurable filtering pipelines, but their flexibility is limited to predefined components and they lack integrated normalisation. These limitations highlight the need for approaches that are language-agnostic and modular, making them easy to update and customise.

Sentence embeddings have become a central component of multilingual corpus filtering, enabling semantic comparison of sentence pairs to ensure alignment without relying on language-specific resources. Early approaches such as LASER (Artetxe and Schwenk, 2019) and LaBSE (Feng et al., 2022) demonstrated strong cross-lingual alignment across hundreds of languages, allowing for effective detection of misaligned or non-parallel segments. More recent work, including the SONAR (Duquenne et al., 2023) embedding suite, further improves multilingual coverage and representational consistency across typologically diverse languages. Embedding-based filtering has become a robust alternative to supervised classification, particularly in low-resource or massively multilingual settings, and is increasingly used in large-scale shared tasks on parallel data curation (Sloto et al., 2023).

Incorrect language labelling is another common source of noise (Khayrallah and Koehn, 2018). Many automatic language identification systems report high accuracy on standard benchmarks, but often only cover a narrow set of high-resource languages, such as Idiomata Cognitor (Galiano-Jiménez et al., 2024), or degrade on shorter, context-free web sentences. FastText (Joulin et al., 2017) is widely used but shows reduced accuracy in such conditions. More recent multilingual systems, such as GlotLID (Kargaran et al., 2023), extend coverage to more than 2000 language varieties and demonstrate improved robustness under low resource and short context conditions, making them particularly suitable for large-scale corpus cleaning.

Beyond alignment and language identification, deduplication and normalisation are essential components of effective preprocessing pipelines. Deduplication reduces redundancy and can improve downstream model performance by mitigating memorised text (Lee et al., 2022). Normalisation and "restorative" cleaning steps, as implemented in tools such as Bifixer (Ramírez-Sánchez et al., 2020) and highlighted in WMT 2023 shared tasks (Sloto et al., 2023), standardize sentence formatting and remove systematic inconsistencies. ParaCLEAN integrates these steps into a modular workflow, allowing individual components to be evaluated independently while improving overall corpus quality.

Against this background, ParaCLEAN contributes a modular pipeline that integrates multiple levels of filtering and normalisation into a single reproducible workflow. By systematically evaluating individual steps, ParaCLEAN highlights where different cleaning techniques have the most impact, and how they can be combined to improve the quality of parallel corpora.

3. The ParaCLEAN Pipeline

3.1. Design Principles

ParaCLEAN is designed as a modular and configurable workflow for preparing parallel corpora. Its goal is to provide a reproducible sequence of processing steps that can be adapted to different language pairs and data sources. The design emphasises three principles: (1) **modularity**, so that steps can be enabled, disabled, or reordered for different experimental setups; (2) **transparency**, with intermediate outputs and logs available for inspection; and (3) **reproducibility**, by providing default settings that work out of the box while still allowing parametrisation. The pipeline is implemented as a lightweight command-line tool with a simple configuration file. Its structure is intentionally transparent, making it easy for users to integrate alternative embedding or language identification models, add new steps, or adapt existing ones to specific data sources.

Figure 1 illustrates the ParaCLEAN workflow, which proceeds through a sequence of modular stages - language identification, embedding-based alignment, thresholded filtering, deduplication, Bifixer validation, and language-specific normalisation - each described in the following section.

3.2. Processing Stages

We describe each stage in more detail below. Each stage is implemented as an independent module that can be modified or replaced without affecting the rest of the pipeline, supporting straightforward experimentation and extension.

3.2.1. Embeddings

For each sentence pair we compute cosine similarity between the source and target sentence embeddings, yielding a score between 0 and 1. This value is stored as metadata and later used in the filtering stage (Section 3.2.3). Embedding similarity complements language identification by detecting semantically weak or misaligned pairs that may otherwise pass lexical or length-based checks.

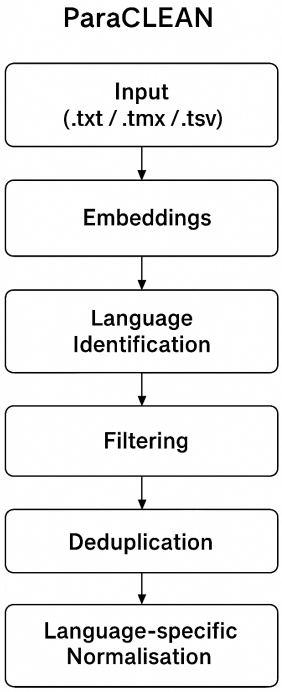

Figure 1: Overview of the ParaCLEAN pipeline. Each stage can be enabled, disabled, or reordered as required.

By default, ParaCLEAN uses LaBSE, chosen for its broad language coverage and permissive licensing. LaBSE is automatically downloaded if no local installation is available, enabling straightforward deployment. For users with access to SONAR, the pipeline can optionally leverage this model by pointing to a local installation.

3.2.2. Language Identification

We apply GlotLID to every segment. For each sentence, GlotLID assigns a probability to each supported language; we retain only the normalised probability of the expected language (e.g. Catalan for the source side, Japanese for the target side). As in the previous step, these probabilities are stored as metadata and passed to later stages.

GlotLID was chosen as the default identifier because it demonstrates superior performance across a wide range of languages, including low-resource ones, compared to other identifiers we evaluated. We tested four language identification systems: Lingua-py, fastText, Idiomata Cognitor, and GlotLID on two datasets of differing quality: FLORES+ (NLLB Team et al., 2024) and WMT24++ (Deutsch et al., 2025). GlotLID achieved the highest identification accuracy on FLORES+ at 91.07% while maintaining strong performance on the noisier WMT24++ set (87.92%), confirming its robustness across both curated and real-world data. Its outputs are both robust and interpretable, enabling flexible use in the filtering stage (Section 3.2.3) where users can set per-language thresholds. All probabilities are logged to support transparency and reproducibility.

3.2.3. Filtering

The filtering stage removes sentence pairs that do not meet user-specified quality thresholds. Each pair is evaluated using the embedding-based alignment score (Section 3.2.1) and the language identification probabilities (Section 3.2.2). The process reads the input file line by line, retaining only those pairs whose alignment and language ID scores exceed the defined thresholds, and writes the filtered sentences to a new output file. This lightweight approach ensures that only semantically and linguistically reliable pairs are preserved for downstream tasks.

3.2.4. Deduplication

ParaCLEAN implements both exact and near-duplicate removal to limit the influence of repeated content and mitigate overfitting. This step is especially important for corpora that aggregate multiple sources, where boilerplate material such as navigation menus, disclaimers, or repeated template text may otherwise dominate the dataset.

Deduplication is performed using a two-step approach. First, each sentence pair is aggressively normalised: text is lowercased, non-alphabetic characters are removed, accents are stripped, and extra whitespace is collapsed. This ensures that minor variations in punctuation, casing, or diacritics do not prevent detection of duplicates. Next, a hash of the normalised source and target sentences is computed. For efficiency, pairs are streamed through an external stable sort by hash, and only the first occurrence of each hash is retained. When multiple pairs share the same hash, a simple ranking heuristic is used to retain the pair with the highest textual content value.

This method efficiently handles very large datasets while ensuring that only unique sentence pairs are preserved for downstream processing. The combination of normalisation, hashing, and streaming sorting enables robust deduplication even in the presence of minor textual variations.

3.2.5. Bilingual Alignment and Text Cleaning (Bifixer)

ParaCLEAN can optionally apply Bifixer, a tool designed to correct common issues in parallel corpora. Bifixer performs multiple text normalisation and cleaning operations, including: fixing mojibake, converting HTML entities, normalising punctuation and spaces, correcting characters from wrong alphabets, fixing orthographic errors for certain lan-

guages, removing empty sentences, and addressing common tokenisation issues. It can also segment long sentences and provide hashes for deduplication or near-duplicate detection.

Unlike other stages, Bifixer requires a separate installation; if it is not installed, this step is skipped automatically. Users can configure which Bifixer features to apply; for integration with ParaCLEAN, we recommend using `–ignore_normalisation` and `–ignore_duplicates`, as the pipeline already performs these operations. Other flags, such as orthography fixes or segmentation, may be applied at the user's discretion depending on the characteristics of the corpus.

3.2.6. Language-Aware Normalisation

ParaCLEAN applies a light, language-aware normalisation to each sentence to improve consistency and reduce trivial noise, while avoiding aggressive transformations that could remove meaningful variation. Each sentence undergoes core steps including Unicode canonicalisation (NFC), trimming of leading and trailing whitespace, collapsing consecutive ASCII spaces, and removing invisible or control characters.

In addition, language-specific normalisation is applied where appropriate. For example, our current pipeline includes more extensive routines for Chinese (simplified) and Japanese, handling common punctuation and script-specific issues. Hindi normalisation additionally addresses numeric and punctuation consistency. For other languages, the normalisation primarily targets quotation marks and minor typographic variation. Minimal normalisation is currently implemented for Arabic.

This light-touch design removes only non-semantic inconsistencies while preserving the diversity of the text. Both the original and normalised versions are preserved in the output file, allowing users to choose whether to work with noisy or cleaned data. The language-specific functions are modular, and contributions extending coverage or sophistication are welcome.

3.3. Usability and Extensibility

ParaCLEAN is open source and lightweight, requiring only a small set of dependencies. Input data in .txt, .tmx, or .tsv format is accepted; non-TSV files are converted to TSV for uniform processing. The pipeline is modular and configured through a YAML file, allowing users to select which processing stages to apply, set thresholds, and control the order of execution. All stages, including filtering, deduplication, language-specific normalisation, and optional Bifixer processing, can be enabled or skipped independently.

During processing, metadata such as cosine similarity and language identification probabilities is added to the input TSV and used in filtering, providing transparency without generating extensive logs. This design facilitates reproducibility, supports integration with existing data preparation workflows, and allows adaptation to diverse language pairs. The modular structure makes it straightforward to extend functionality — for example, by adding new cleaning modules, swapping in different embedding or language identification systems, or integrating ParaCLEAN into larger data preparation workflows.

Each stage produces a new .tsv file. This structure allows users to inspect intermediate outputs, choose what to retain, and resume processing from any step, ensuring transparency and reproducibility while supporting integration into broader data preparation workflows. The modular structure also makes it straightforward to extend functionality—for example, by adding new cleaning modules, replacing embedding or language identification components, or embedding ParaCLEAN within larger pipelines.

To evaluate the impact of individual stages and the pipeline as a whole, we train bilingual Catalan→Japanese models on raw, partially cleaned, and fully cleaned corpora. Details of the experimental setup are provided in Section 4.

4. Experiments

4.1. Overview

To assess the impact of individual processing stages in ParaCLEAN on downstream translation performance, we train 6 bilingual Catalan→Japanese models on data processed with different configurations of the pipeline. These include raw data, the full pipeline (excluding Bifixer), and four ablation configurations omitting one cleaning stage each. Bifixer is excluded from the experimental configurations, as it is an external tool not developed within ParaCLEAN and is itself extensively configurable.

4.2. Data

The training corpus consists of parallel Catalan and Japanese data. We first gathered all available authentically parallel resources with suitable licenses from OPUS (Tiedemann, 2012), totaling approximately 1.7 M sentence pairs.

Given the low-resource nature of this language pair, we expanded the corpus by generating synthetic parallel data. Specifically, we translated the Spanish side of Spanish–Japanese corpora into Catalan using SalamandraTA-7b (Gilabert et al., 2025). This synthetic augmentation increased the

corpus to 25 M sentence pairs. Table 1 summarizes the main corpus statistics. The slightly longer average sentence length in the synthetic Catalan portion likely reflects the relatively verbose nature of the Spanish source texts and the literal translation style of the MT system. As this discrepancy occurs on the source side, it is not expected to significantly affect alignment or model performance.

All corpora included in the final dataset had previously been aligned and filtered by their respective creators or distributors, though with differing tools and criteria. Prior to applying ParaCLEAN, no further preprocessing was performed. A detailed list of all included corpora is provided in Appendix A.

4.3. Pipeline Variants

We evaluate six pipeline configurations, summarised in Table 2. Run 1 corresponds to the raw baseline with only minimal preprocessing, while Run 2 applies the complete ParaCLEAN pipeline (excluding Bifixer). Runs 3-6 are ablation settings in which one component is removed at a time in order to isolate its contribution. Specifically, we examine the effect of excluding (i) embedding-based filtering, (ii) the language identification step, (iii) deduplication, and (iv) normalisation.

In all runs where these components are active, we apply a LaBSE similarity threshold of 0.75 and a GlotLID confidence threshold of 0.5, chosen based on preliminary experiments to balance coverage and precision. Normalisation, when applied, is limited to the target (Japanese) side; the source (Catalan) side is intentionally left unnormalised to preserve natural noise and variability, promoting robustness in the trained translation models.

The corpus size reductions reported in Table 2 illustrate the cumulative effect of each stage. The raw corpus (Run 1) contains 24.1 M pairs, while the complete pipeline (Run 2) retains 10.2 M, reflecting the strong selectivity of the filtering and deduplication stages. Intermediate ablations yield proportionally larger subsets, indicating that all components contribute meaningfully to noise reduction. Notably, removing either the embedding-based or language-identification filter (Runs 3–4) results in markedly larger corpora, suggesting that these steps target distinct sources of noise.

4.4. Training

Our neural machine translation systems are trained using Fairseq (Ott et al., 2019). We build Transformer-based encoder-decoder models (Vaswani et al., 2017) with 24 encoder layers and 6 decoder layers, each with pre-layer normalisation.

For data preparation, we train a single SentencePiece model (Kudo and Richardson, 2018) per language (one for the source and one for the target) on the raw corpus, rather than separately for each experimental setting. This approach ensures consistent segmentation across pipeline variants and prevents tokenisation differences from influencing model comparisons. Each vocabulary has 50,000 subword units. Preprocessing is handled through Fairseq's lazy dataset implementation.

Models are optimised using Adam (Kingma and Ba, 2015) with β=(0.9, 0.98). An inverse square-root learning rate scheduler with 4000 warm-up steps, and a peak learning rate of 0.001. We apply dropout at 0.1, weight decay of 0.0003, and label-smoothed cross-entropy with a smoothing factor of 0.1. Training uses a maximum of 2400 tokens per batch per GPU, gradient accumulation with an update frequency of 25, and mixed-precision (fp16) training.

All models are trained with identical architectures and hyperparameters, differing only in the input data derived from the various pipeline configurations. Training proceeds until convergence on the validation set, with early stopping after two consecutive non-improving validation runs.

5. Results and Analysis

5.1. Evaluation Overview

We evaluate all model variants on the FLORES+ devtest set. Translation quality is measured with BLEU (Papineni et al., 2002), COMET (Rei et al., 2020), TER (Snover et al., 2006), and chrF (Popović, 2015). All models share an identical architecture, training setup, and hyperparameters; only the input data differ according to the pipeline configuration. This ensures that any observed variation in translation quality can be attributed solely to the data preparation stages.

Ablation runs are compared against the full ParaCLEAN configuration to isolate the contribution of each component. In addition to system-level metrics, we analyse the filtering stages directly - examining the overlap, selectivity, and data characteristics of the embedding-based and language-identification filters (see Section 5.3). We also analyse the script composition on the Japanese side to quantify noise reduction effects. These diagnostics clarify how filtering decisions influence both dataset composition and model behaviour.

5.2. Translation Quality

Table 3 summarises the performance of all six experimental runs, ranging from training on raw data to successive ablations of the full pipeline configuration.

Applying the full ParaCLEAN pipeline leads to a substantial improvement over training on raw data, with BLEU increasing from 3.3 to 25.7, COMET

Statistic	Catalan		Japanese (Collected)
	Collected	Synthetic from ES	
Sentence pairs	1,698,452	24,090,700	25,786,935
Average sentence length (chars)	57.10	60.00	29.56
Median sentence length (chars)	29	41	20
Vocabulary size		64,108	61,012

Table 1: Corpus statistics for Catalan (collected and synthetic from ES) and Japanese (collected). The synthetic Catalan side was generated by translating collected Spanish sentences; both Catalan portions share a common vocabulary.

Run	Embeddings	Language ID	Filtering	Deduplication	Normalisation	Sentence pairs (M)
1	×	×	×	×	×	24.09
2	✓	✓	✓	✓	✓	10.20
3	×	✓	✓	✓	✓	19.69
4	✓	×	✓	✓	✓	11.33
5	✓	✓	✓	×	✓	11.81
6	✓	✓	✓	✓	×	10.20

Table 2: Pipeline configurations and resulting corpus sizes after filtering. Sentence counts are shown in millions (rounded to two decimals). "✓" denotes that a stage is active; "×" indicates it was disabled. Counts reflect application of the stated thresholds (LaBSE ≥ 0.75, GlotLID ≥ 0.5 where applicable).

Run	**BLEU↑**	**COMET↑**	**TER↓**	**chrF↑**
1	3.34	0.477	199.26	7.16
2	25.69	0.854	131.54	32.61
3	25.85	0.851	132.93	32.92
4	**25.86**	0.851	131.73	32.85
5	25.76	0.851	133.49	32.82
6	25.78	0.854	**131.08**	**33.00**
MADLAD 7B	24.09	**0.864**	207.68	31.55
NLLB 3B	16.59	0.852	306.38	26.65

Table 3: Catalan→Japanese translation performance on the FLORES+ devtest set across pipeline ablation runs.

from 0.48 to 0.85, and TER dropping by more than 65 points. These gains confirm that the quality and consistency of the training data, rather than its raw size, are the dominant factors in low-resource translation performance.

Among the ablation runs (3–6), differences in the standard metrics are small-typically within 0.1 BLEU or COMET-indicating that no single component is solely responsible for the overall improvement. Instead, the benefit arises from the joint effect of multiple cleaning stages. For example, removing either the embeddings or language-identification filter (Runs 3–4) yields almost identical scores, suggesting that the two filters target partly overlapping but complementary forms of noise. Likewise, disabling deduplication (Run 5) or normalisation (Run 6) has minimal effect on aggregate metrics, though the latter shows slightly better consistency across evaluation criteria.

For reference, we also evaluate two large multilingual models, MADLAD-7B and NLLB-3B. Our bilingual models outperform both baselines across most metrics, with the exception of COMET for MADLAD-7B—a considerably larger model—highlighting that carefully curated bilingual data can yield competitive results even against substantially larger multilingual systems.

Taken together, these results show that ParaCLEAN's gains stem from cumulative data refinement rather than any single decisive filter. The minor metric fluctuations between ablations further suggest that once a sufficient baseline of data quality is achieved, remaining noise sources exert only a marginal influence on average test performance-though they may still affect robustness and rare-case generalisation, explored below.

5.3. Filter Overlap and Selectivity

To understand how the embeddings-based and language-identification (langid) filters interact, we examined sentence-level overlap. A Jaccard similarity of 0.50 indicates that each filter captures distinct subsets of noisy sentences, providing complementary coverage rather than redundancy. Figure 2 visualizes the relative sizes of the sets and their intersection.

The langid stage removes overtly noisy or cross-lingual text, while the embeddings filter retains semantically valid pairs that langid would exclude, such as Romanised Japanese or code-switched content. The combined setup therefore favours higher precision at the expense of some recall.

Cosine-similarity statistics reinforce this distinction. Mean similarity rises from 0.72 in the full dataset to 0.86 among pairs kept by the embed-

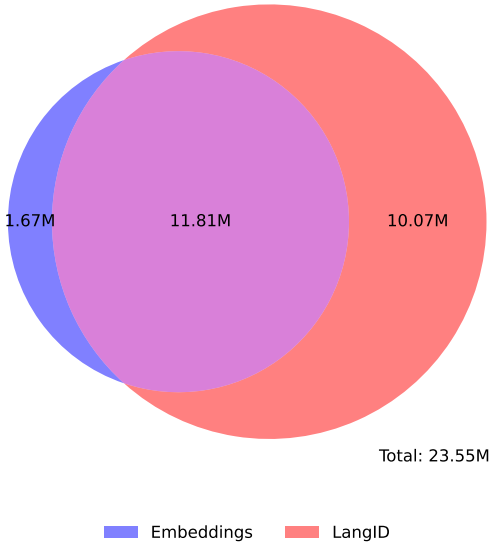

Figure 2: Venn diagram showing the overlap of sentences retained by the embeddings and langid filters.

dings filter, isolating semantically well-aligned examples. In contrast, sentences kept only by langid show lower similarity, reflecting a broader tolerance for misaligned pairs (Table 4).

Subset	Mean	Std	Min	Max
Raw data	0.716	0.185	-0.49	1.00
Embeddings-kept	**0.864**	0.069	0.75	1.00
LangID-kept	0.580	0.148	-0.47	0.749

Table 4: Cosine similarity statistics across subsets. Higher values indicate stronger semantic alignment between source and target.

Side	Mean	Median	Min	Max
L1 (source)	29.1	0.27	0.001	100
L2 (target)	25.5	0.05	0.001	100

Table 5: Language-identification probabilities for sentences kept only by the embeddings filter. Lower probabilities indicate uncertain or misclassified language detection.

Language-identification probabilities for the embeddings-only subset vary widely (Table 5), often low due to named entities, Romanised text, or mixed scripts. This shows that the two filters detect different noise types.

Despite these complementary behaviours, their downstream impact is limited: translation quality remains essentially unchanged when either filter is removed (see Table 3). This suggests that both capture overlapping forms of noise and that the main gains from ParaCLEAN arise from the collective effect of the full pipeline rather than from either filtering stage in isolation.

5.4. Latin Script Proportion in Data and Model Outputs

To quantify cross-script contamination and the effects of noise reduction, we measured the proportion of Latin characters on the target-side Japanese training data and on the corresponding model outputs, against FLORES+ devtest targets. Character types were identified via Unicode blocks and normalised by total character count.

Training data. In the raw corpus, 25.6% of characters on the Japanese side were in Latin script. After full cleaning (Run 2), this proportion dropped to 20.6%, corresponding to 83.3 million Latin characters, effectively minimising Latin content. Disabling language identification (Run 4) yielded the highest residual Latin share (24.1%), confirming that language ID plays a central role in filtering mixed-script material (Table 6).

Run	JP (%)	Latin (%)	Other (%)	Total Chars (M)	Latin Chars (M)
Raw	68.84	**25.63**	5.53	786.9	**201.5**
2 (full)	74.09	20.62	5.29	404.0	**83.3**
3 (–emb)	75.10	19.11	5.78	639.5	122.2
4 (–langid)	70.74	**24.12**	5.14	431.8	104.1
5 (–dedup)	73.32	21.36	5.31	451.9	96.5
6 (–norm)	74.08	20.59	5.34	404.1	83.3

Table 6: Character composition of Japanese-side training data where JP = Hiragana + Katakana + CJK (kanji). The full pipeline produces a smaller, cleaner dataset with a lower relative share of Latin characters, while removing language identification causes mixed-script content to increase.

Model outputs. On FLORES+ devtest, the raw model (Run 1) produced 15.8% Latin characters in Japanese output, versus 8.3% for the fully cleaned model (Run 2). The LangID ablation (Run 4) again showed the highest leakage (8.45%).

Figure 3: Proportion of Latin characters in Japanese model outputs across pipeline variants. Language identification (Run 4) is the key factor preventing script leakage.

The numbers indicate that ParaCLEAN substantially reduces Latin-script contamination in both training data and model outputs. The consistency between data-level and model-level reductions indicates that script composition in training data can

influence generation behaviour. Improving script consistency is especially relevant for downstream readability and evaluation in multilingual settings.

5.5. Robustness on Numbers/Units/Dates/Ranges (NUDR)

Robustness is evaluated using a 200-item Numbers, Units, Dates, and Ranges (NUDR) stress set [2] designed to test resilience to small formatting perturbations. Each source item appears in a CLEAN and a NOISY form differing only by controlled surface changes such as separator swaps or spacing. Model outputs are checked with a regular-expression acceptor under two regimes: (i) strict canonical matching and (ii) lenient matching that allows common variants. Before matching, hypotheses are normalised with NFKC.

Robustness is summarised as

$$\Delta = \mathrm{Acc}_{\mathrm{clean}} - \mathrm{Acc}_{\mathrm{noisy}}$$

where smaller Δ indicates less degradation under perturbation (i.e., greater robustness).

Strict matching. Under canonical formatting, the full pipeline (Run 2) shows the smallest degradation under noise and the highest noisy accuracy (Table 7). Absolute accuracies are intentionally low due to adversarial perturbations.

Run	Strict			Lenient		
	Acc (clean)	Acc (noisy)	Δ	Acc (clean)	Acc (noisy)	Δ
2 (full)	0.28	**0.13**	**0.16**	0.35	0.13	0.22
3 (no emb)	0.31	0.09	0.23	0.36	0.09	0.28
4 (no langid)	0.32	0.10	0.22	0.37	0.14	0.24
5 (no dedup)	0.31	0.12	0.20	0.35	**0.15**	**0.21**
6 (no norm)	**0.35**	0.10	0.26	**0.39**	0.13	0.26

Table 7: NUDR (N=200). Comparison of strict vs. lenient acceptors. $\Delta = \mathrm{Acc}_{\mathrm{clean}} - \mathrm{Acc}_{\mathrm{noisy}}$; smaller is better.

Lenient matching. When the acceptor allows common variants (e.g., "yen"/¥, en-dash/∼, km/"kilo", optional interword spacing), Run 5 trends best, but gaps remain small.

Overall, the full pipeline best preserves numeric and symbolic fidelity under noise, underscoring the importance of normalisation and preprocessing for robust NUDR handling; however, given the small sample size and low absolute accuracies, these robustness trends should be interpreted cautiously.

[2] https://github.com/langtech-bsc/ParaCLEAN/tree/main/tests

6. Interpretation and Discussion

The ParaCLEAN pipeline yields substantial gains over raw data (BLEU 3.34→25.69; COMET 0.477→0.854), confirming the decisive role of corpus quality in low-resource translation. Ablations show that no single component drives the improvement: removing embedding-based or language-identification filters changes scores only marginally, while deduplication consistently reduces redundancy. The effect of normalisation on automatic metrics is minor and not entirely monotonic (e.g., Run 6 shows slightly higher BLEU and chrF without it), but it serves a complementary role in harmonising punctuation and script usage, which may benefit linguistic fidelity in ways not fully captured by BLEU, COMET, or TER. Together, the stages act synergistically-each targeting distinct noise types to produce a cleaner, semantically aligned core corpus.

Once this clean core is established, further data mainly adds stylistic variation with diminishing returns, explaining the tight clustering of ablation results. This aligns with our data-level analysis: embedding and language-ID filters overlap substantially (Jaccard ≈ 0.50) yet differ in the noise they remove, accounting for their complementary effects. Beyond aggregate metrics, stress and script analyses reveal that cleaning also shapes robustness and linguistic fidelity: the full pipeline is most stable under strict acceptance, while deduplication slightly reduces tolerance to surface-form variation.

7. Conclusion

We presented ParaCLEAN, a modular, reproducible workflow for cleaning parallel corpora. Applied to a low-resource and typologically distant pair (Catalan→Japanese), it delivers large quality gains while demonstrating that improvements arise from the combined action of complementary filters rather than any single stage. Although tuned for this setting, ParaCLEAN's components - language identification, embedding-based filtering, deduplication, and language-aware normalisation - are fully configurable: thresholds, models (e.g., LaBSE or SONAR), and language-specific rules can be easily adapted or omitted. This flexibility makes ParaCLEAN applicable across diverse data conditions, offering both a practical cleaning toolkit and an interpretive framework for understanding how data quality drives translation performance.

8. Limitations and Ethical Considerations

Our experiments focus exclusively on Catalan–Japanese, a low-resource and typologically diverse

pair. While this setting highlights the importance of cleaning noisy corpora, results may not directly generalise to other language pairs. Future work should therefore validate the findings across a broader range of languages and domains.

A second limitation is the reliance on synthetic parallel data, generated by translating Spanish into Catalan. Synthetic corpora can introduce translationese artifacts, reinforce systematic errors of the generating model, and potentially bias evaluation outcomes. Although we report experiments transparently, downstream applications should be aware of these biases.

Finally, ParaCLEAN performs filtering based on thresholds for language identification and embedding similarity. These thresholds inevitably involve trade-offs between precision and recall of retained sentence pairs, and different settings may lead to different conclusions. We encourage replication with alternative thresholds to confirm the robustness of our findings.

Acknowledgements

This work/research has been promoted and financed by the Government of Catalonia through the Aina project.

This work is funded by the Ministerio para la Transformación Digital y de la Función Pública and Plan de Recuperación, Transformación y Resiliencia - Funded by EU – NextGenerationEU within the framework of the project Desarrollo Modelos ALIA.

9. Bibliographical References

Haluk Açarçiçek, Talha Çolakoğlu, Pınar Ece Aktan Hatipoğlu, Chong Hsuan Huang, and Wei Peng. 2020. Filtering noisy parallel corpus using transformers with proxy task learning. In *Proceedings of the Fifth Conference on Machine Translation*, pages 940–946, Online. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, page 3197–3203. Association for Computational Linguistics.

Mikko Aulamo, Sami Virpioja, and Jörg Tiedemann. 2020. OpusFilter: A configurable parallel corpus filtering toolbox. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 150–156, Online. Association for Computational Linguistics.

Daniel Deutsch, Eleftheria Briakou, Isaac Caswell, Mara Finkelstein, Rebecca Galor, Juraj Juraska, Geza Kovacs, Alison Lui, Ricardo Rei, Jason Riesa, Shruti Rijhwani, Parker Riley, Elizabeth Salesky, Firas Trabelsi, Stephanie Winkler, Biao Zhang, and Markus Freitag. 2025. Wmt24++: Expanding the language coverage of wmt24 to 55 languages dialects.

Paul-Ambroise Duquenne, Holger Schwenk, and Benoît Sagot. 2023. Sonar: Sentence-level multimodal and language-agnostic representations.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Aarón Galiano-Jiménez, Felipe Sánchez-Martínez, and Juan Antonio Pérez-Ortiz. 2024. Idiomata cognitor.

Javier Garcia Gilabert, Xixian Liao, Severino Da Dalt, Ella Bohman, Audrey Mash, Francesca De Luca Fornaciari, Irene Baucells, Joan Llop, Miguel Claramunt Argote, Carlos Escolano, and Maite Melero. 2025. From salamandra to salamandrata: Bsc submission for wmt25 general machine translation shared task.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pages 427–431.

Amir Hossein Kargaran, Ayyoob Imani, François Yvon, and Hinrich Schütze. 2023. GlotLID: Language identification for low-resource languages. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. *arXiv preprint arXiv:1805.12282*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. 2020. Findings of the wmt 2020 shared task on parallel corpus filtering and alignment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 726–742.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the wmt 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L Forcada. 2018. Findings of the wmt 2018 shared task on parallel corpus filtering. In *EMNLP 2018 Third Conference on Machine Translation (WMT18)*, pages 726–739. Association for Computational Linguistics.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan Van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, et al. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better.

Jun Lu, Xin Ge, Yangbin Shi, and Yuqi Zhang. 2020. Alibaba submission to the wmt20 parallel corpus filtering task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 979–984, Online. Association for Computational Linguistics.

Alexandra Sasha Luccioni and Joseph D Viviano. 2021. What's in the box? a preliminary analysis of undesirable content in the common crawl corpus. *arXiv preprint arXiv:2105.02732*.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2024. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation (WMT)*, pages 392–395. Association for Computational Linguistics.

Gema Ramírez-Sánchez, Jaume Zaragoza-Bernabeu, Marta Bañón, and Sergio Ortiz-Rojas. 2020. Bifixer and bicleaner: two open-source tools to clean your parallel data. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 291–298, Lisboa, Portugal. European Association for Machine Translation.

Ricardo Rei, Ana Farinha, Alon Lavie, and André F. T. Martins. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702. Association for Computational Linguistics.

Steve Sloto, Brian Thompson, Huda Khayrallah, Tobias Domhan, Thamme Gowda, and Philipp Koehn. 2023. Findings of the WMT 2023 shared task on parallel data curation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 95–102, Singapore. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 223–231. Association for Machine Translation in the Americas.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Rik Van Noord, Miquel Esplà-Gomis, Mălina Chichirău, Gema Ramírez-Sánchez, and Antonio Toral. 2025. Quality beyond a glance: Revealing large quality differences between web-crawled parallel corpora. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1824–1838.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Jaume Zaragoza-Bernabeu, Gema Ramírez-Sánchez, Marta Bañón, and Sergio Ortiz Rojas. 2022. Bicleaner ai: Bicleaner goes neural. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 824–831.

10. Appendix A: Parallel Data Sources

Dataset	Licence	Sentence pairs
Catalan–Japanese (authentic)		
KDE4 (v2)	CC BY 4.0	120,969
MultiCCAligned (v1.1)	CC BY 4.0†	804,536
OpenSubtitles (v2024)	CC BY 4.0	116,270
WikiMatrix (v1)	CC BY-SA 4.0	103,899
XLEnt (v1.2)	CC BY 4.0	552,778
Total		**1,698,452**
Spanish–Japanese (synthetic, es→ca)		
MultiCCAligned (v1.1)	CC BY 4.0†	7,057,829
NLLB (v1)	ODC-BY	15,402,228
OpenSubtitles (v2024)	CC BY 4.0	1,922,477
WikiMatrix (v1)	CC BY-SA 4.0	219,261
Total		**24,601,795**

Table 8: Parallel data sources and sizes for Catalan–Japanese and Spanish–Japanese datasets. All corpora were obtained from OPUS. The Spanish side of synthetic corpora was translated into Catalan using SalamandraTA-7b.

terms of agreement.